Information systems flexibility

Spoor, E.; Boogaard, M.

1993

Faculteit der Economische Wetenschappen en Econometrie

Serie Research Memoranda

Information Systems Flexibility: A Conceptual Framework

E.R.K. Spoor
M. Boogaard

Research Memoranda 1993-59
December 1993

vrije Universiteit amsterdam

Information Systems Flexibility: A Conceptual Framework

E.R.K. Spoor and M. Boogaard

Abstract

The information systems field has pursued several different objectives such as productivity, quality, and integration as the primary focus of research and practice. The latest seems to be flexibility. In many fields, flexibility is seen as the ultimate philosophy of the 1990s. Although Ackoff identified the need for flexible information systems as early as 1967, the importance of flexibility has only recently begun to be acknowledged. Unfortunately, problems develop because flexibility turns out to be difficult to define and to operationalize, and because flexibility measurement criteria are hard to identify. Furthermore, flexibility is frequently used as a synonym for issues such as portability, reusability, and maintainability. The problems that pertain to flexibility are general; they are not unique to the field of information systems. In this paper, a conceptual framework is introduced to understand information systems flexibility. The potential of the framework is exemplified by applying it to three contemporary developments that claim to introduce or increase the flexibility of information systems, i.e., object orientation, CASE technology, and relational systems.

Key Words

Information Systems, Flexibility, Information Systems Flexibility Framework, Object Orientation, CASE Technology, Relational Systems

1. INTRODUCTION

The rapid pace of change in our modern post-industrial environment creates significant instability and uncertainty for organizations especially for large ones (see, e.g., Volberda 1992; Cameron 1986). This external environment generates new requirements to both the products and services, and the organization and management of firms. It is generally accepted that flexibility is one of the key features of effective organization in a turbulent environment. For instance, Steers' review of 17 multivariate models of organizational effectiveness in terms of their primary evaluation criteria, reveals that flexibility is most often mentioned (1975). In this sense, flexibility refers to the capability to adapt to changing conditions that ensures the continuity of organizations. The acknowledgment of the importance of flexibility, has resulted in numerous theoretical studies on the subject of organizational flexibility since the early 70s (see, e.g., Volberda 1992; Eppink 1978; Ackoff 1977). Not only in organization and management science but also in other fields, for example, production systems (see, e.g., Crowe 1992; Edquist and Jacobsson 1988; Zelenovic 1982), logistics (see, e.g., Geraerds and Igel 1989), and employment (see, e.g., Standing 1991), flexibility is considered the answer to threats and opportunities that are brought about by the increasing dynamics and complexity of environments. This state of affairs seems to support the case of those who say that flexibility is one of the common-sense principles of our time (see Perez 1985) or those who say that flexibility is the ultimate philosophy of the 1990s (see Crowe 1992).

It is interesting to observe that the studies in these different fields share the similar problems pertaining to the phenomenon flexibility. The most fundamental problem, and also generally acknowledged as such, concerns the definition of flexibility. The term flexibility describes a vague conception that can be interpreted and applied in various ways. The following citation of Gerwin and Leung (1980) may serve as a good example of this.

> "Flexibility is usually considered the most important attribute of a Flexible Manufacturing System. Therefore, it is surprising that a great deal of ambiguity surrounds the term. The difficulty with flexibility arises from the existence of several alternative meanings, and the inability to quantify its benefits."

Research on the topic of flexibility has revealed many different opinions resulting in a 'mish-mash' of definitions and meanings (see, e.g., Volberda 1992; Kickert 1982; Zelenovic 1982; Eppink 1978; Scott 1965). The definitions differ in many ways, for example, they delineate different aspects, and they often concern a specific context. The variety in definitions and use is caused by the fact that flexibility, like other concepts such as quality (see Pirsig 1981; Boehm et al. 1978), is difficult to define, operationalize, and measure in a straight-forward manner. It turns out to be a complex, multidimensional and varied concept as, for example, Crowe (1992) illustrates. Notwithstanding these difficulties, renewed efforts have recently been made to capture the meaning and purpose of the word (see, e.g., Volberda 1992). The limitations of these contributions may be traced to their lack of a fundamental theoretical basis.

One of the fields in which flexibility is being recognized as increasingly important is that of information systems. Because the environment of an information system is the organization and because information systems mirror aspects of organizations, it is evident that increasing organizational flexibility will have consequences for information systems. For instance, Frazelle (1986) argues that in order for organizations to compete labor, management, facilities, manufacturing processes, and information systems must all be able to adapt to new circumstances quickly and inexpensively. A similar line of reasoning can be found in Allen and Boynton (1991) who say that today organizations need the most flexible information systems they can find.

Several authors, for example, Allen and Boynton (1991), Fitzgerald (1990), Rochester (1989), Land (1982), and Ackoff (1967), explicitly refer to the subject of information systems flexibility. None of these authors, however, provide an operational and comprehensive definition of the notion. As a consequence, the meaning of information systems flexibility relies on an intuitive understanding of the term. Hence, the field of information systems also encounters the problems faced in other fields as described above which results in miscommunication. The different levels and perspectives at which information systems flexibility is considered by these authors and the homonymous use of the term flexibility to address issues such as portability, reusability, and maintainability may easily result in misunderstandings. Furthermore, the ambiguity also creates difficulties in assessing the precise value of flexibility enhancing proposals.

For instance, questions like "Do object oriented supporters and CASE technology supporters mean the same thing if they talk about flexibility?" are difficult to answer appropriately without a clear definition or demarcation of the concept. In addition, clarity of the term is also a prerequisite for the operationalization of flexibility and the measurement of the degree of flexibility of information systems.

This paper attempts to sketch a direction that seems fruitful to follow if we are to achieve a theoretical structure that can be helpful in guiding research on and thinking about information systems flexibility and flexibility in general. The theoretical structure introduced in this paper is a reasonably coherent intellectual framework which integrates existing knowledge from various disciplines, e.g., organizational theory, and production systems. Existing knowledge pertaining to flexibility is rather diverse and difficult to relate because of the lack of a theoretical basis underlying the different contributions. The premise of this paper is that the linguistic grammar theory may serve as a conceptual bridge between the different disciplines. As a whole, the paper is intended as food for thought and conceived mainly as a contribution for opening new paths in thinking about (information systems) flexibility.

The paper presents and illustrates the value of a conceptual framework for information systems flexibility. The framework can be used to understand and compare past and present developments in the information systems field with respect to their contribution to flexibility. The principle of the framework, grasped from grammar theory, relies on the view that a situation in which flexibility is involved can be seen as a scenario with entities that play different roles. The paper identifies a limited number of roles that are common in different information systems flexibility scenarios. These roles constitute a framework that supports understanding and enables comparison of different scenarios. The framework does not constitute a definition of information systems flexibility neither does it discuss the operationalization of flexibility nor the measurement of the degree of flexibility in full depth. Instead, it enables a classification of flexibility in such a way that for each class a separate definition can be created if necessary.

In the following sections, the paper addresses the problems of information systems flexibility in more detail. Furthermore, two real-life examples will be discussed. Next, Section 3 explains the underlying rationale of the conceptual framework. A separate section (Section 4) is devoted to the construction of the framework. One of the possibilities of the framework is illuminated in Section 5. Successively, three flexibility scenarios of contemporary developments in the information systems field are given, i.e., relational systems, object orientation, and CASE technology. The paper concludes with some remarks on different applications of the framework and provides some suggestions for further research.

2. INFORMATION SYSTEMS FLEXIBILITY

In order to provide an overview of the relevant information systems literature concerning flexibility, it is useful to differentiate several perspectives. At least three perspectives on information systems flexibility can be distinguished: (1) as objective of the application of IT, (2) as natural trajectory of IT developments, and (3) as feature of information systems. These perspectives are briefly described in this section.

Some authors argue that one of the objectives of organizations to apply IT has been to enhance competitive advantage through increased flexibility (see, e.g., Crowe 1992; Geraerds and Igel 1989; Piore and Sabel 1984). In this perspective, the objective of information systems is to provide fast and widespread availability of information which is necessary to react quickly and to reduce uncertainty. Furthermore, organizations endeavour to achieve higher productivity by means of information systems. However, the impact of the application of IT on organizations has a dual character seen from a flexibility point of view. Darnton and Giacoletto (1989) formulate this two-edged effect as follows.

> "IT has the potential either to lock an enterprise into a fixed organizational structure and way of doing business, or to unlock a whole range of possibilities through very flexible and adaptive processes to support different ways of meeting enterprise objectives."

The second perspective considers flexibility as natural trajectory of IT developments. In several articles on technological advance, it is argued that the direction of technological advance follows some inner logic of its own, i.e., a so-called natural trajectory (see, e.g., Dosi 1982; Nelson and Winter 1977). Especially in industries where technological advance is very fast, advances seem to succeed each other in a way that appears somewhat 'inevitable' but certainly do not follow a sound (economic) model (see Nelson and Winter 1977). The field of IT is often characterized as changing extremely fast and is also typified by replacing theories, concepts, and technological advances rather randomly. Generally, natural trajectories depend on the technology considered, or more broadly the 'technological regime' regarded. A technological regime implicitly defines the potentials, constraints, and not yet exploited opportunities. Therefore, the technological regime heads the attention to certain directions in which progress is possible. In conclusion, the particular technological regime both defines the boundaries as well as the trajectories to those boundaries.

The history of IT shows technological advances that, albeit implicitly, claim to introduce or increase flexibility at different levels. Figure 1 illustrates the progression of IT by means of some developments in which the intrinsic drive, among others, seems to be flexibility.

Level	Development
Systems Development Methodologies	Chaotic ► Linear ► Prototyping
Database Management Systems	Flat files ► Hierarchic ► Network ► Relational
Programming Languages	0/1 ► Assembler ► 3GL ► 4GL
Programming Style	"Spaghetti" ► Structured ► Modular ► Object oriented
Systems Architecture	Centralized ► Decentralized

Figure 1. Progression in some IT Developments

It should be noted that the exemplary list of developments in Figure 1 is very rudimentary and is included for illustration only. Nevertheless, altogether the developments suggest that flexibility can be considered one of the natural trajectories within the information systems realm. Support of this perspective can be found in, for example, Perez (1985), and Piore and Sabel (1984).

A third perspective that can be found in the literature, is the point of view that envisions flexibility as a necessary but yet lacking feature of information systems. The following overview provides a summary of some relevant contributions with respect to this perspective.

Already in 1967, Ackoff arrives at the conclusion that an information system should be designed to be flexible and adaptive. Land (1982) observes that information systems are inherently less flexible than people despite the general opinion that people are conservative, dislike innovation, and react adversely to the possibility of change. Therefore, information system users and professionals have looked for methods to improve this state of affairs. Land devotes most of his attention to one of the proposed methods, i.e., improving the designer's model of the world in which the information system will have to operate, and to the possibilities of a future analysis. He concludes that such an analysis is necessary in order to build information systems with a certain extent of flexibility. Fitzgerald (1990) elaborates on the subject of the need for future analysis. He proposes a technique of flexibility analysis and also indicates the benefits and practicality of the technique of flexibility analysis. Rochester (1989) reviews three approaches for building more flexible systems, i.e., designing for change (zero-maintenance systems [see Ligezinski 1988]), automating application development (CASE technology), and aiming for reusability (object orientation). In addition, he distinguishes three points of view on information systems flexibility. First, the business view of information systems flexibility concerns the ability to develop or adjust information systems in accordance to corporate changes. Next, the user's view of information systems flexibility implies both intuitive and tailorable user interfaces. Third, the information systems department view of information systems flexibility denotes a number of things, for instance, portability, connectivity, and maintainability. In a MISQ article of December 1991, Allen and Boynton address the dual challenge for organizations of speed and flexibility on the one hand, and low cost and efficiency on the other hand. They discuss two architectural solutions to the problem, and the advantages and disadvantages of each.

The difficulty with the above perspectives is the diversity of opinions about what information systems flexibility is or how it is understood. Two practical examples may illustrate more specifically how diverse these opinions can be. The first concerns a corporate financial information system for a Dutch multinational. The system primarily contains information about subsidiary companies and their balance sheets and income statements (both consolidated and individual figures). The relevant financial data items of the subsidiary companies differ from one another and vary significantly from year to year. An information system developed using traditional methods would require substantial structural database alterations and application program maintenance in this sort of situation. Thus, the system would be experienced as highly inflexible. The information system was therefore deliberately designed and built for change. All components (e.g., structures and processes) subject to potential change were modelled to be instances of some meta database. With this meta database the end-users were able to change the information system to a large extent without the help of maintenance programmers. Flexibility was built into the system. A more comprehensive description of this information system can be found in Spoor (1992).

The second example involves the degree of flexibility that proceeds from the decision to select a non-proprietary 4GL development environment. A large Dutch organization chose such a tool as the corporate standard for the development of large interactive applications for two reasons: (1) independence from suppliers of (relational) database management systems and hardware platforms, and (2) the resulting portability of implemented applications over a variety of hardware platforms, flat files and database management systems. Thus, whether or not it was based on the right arguments, the organization chose the non-proprietary tool because it would increase information systems flexibility.

3. HOW TO DEFINE INFORMATION SYSTEMS FLEXIBILITY?

Although flexibility is a popular phenomenon in a number of different fields of interest today, an in-depth elaboration of the term itself is rare (exceptions are, e.g., Kickert 1982, Volberda 1992). This is particularly the case with information systems flexibility where there are at least three distinct approaches. The first is to seek one overall definition like the one given by Scott (1965): "... the ability to adjust or adapt to change." Similar definitions can be found in Ansoff (1978), Eppink (1978), and Volberda (1992). Because of the versatile nature of the phenomenon, however, applying such an approach in the area of information systems will only result in a broad definition with hardly any operational value. Its contribution to the understanding of information systems flexibility would be small.

The second approach is to find an ad hoc definition of flexibility suitable for a specific purpose. For example, the definition of Veldwijk (1993) "... the extent to which an information system can be changed without need for re-programming, re-testing, re-debugging and re-documenting" is very narrow and dependent on a specific context.

Another example can be found in Crowe (1992): "... the ability to make substantial changes in schedules and volumes for existing products and to handle frequent product revisions and introductions." Although definitions of this kind do have operational value, their application is quite limited and, therefore, they provide no sound basis to compare and classify various interpretations of information systems flexibility.

The third approach, which will also be the one adopted in this paper, is to construct a flexibility framework instead of seeking for a general or a specific definition. The underlying idea is that flexibility has several dimensions, each of which is a measure for a particular aspect. If these dimensions can be identified and their values distinguished, then a framework can be constructed that is more general than a narrow contextual definition and less superficial than an overall one. The problem is, however, finding these dimensions and values. Looking to organizational research again, there are various different opinions about which dimensions and values are important. For instance, Eppink (1978) identifies three dimensions one of which is the type of flexibility that can be either operational, competitive or strategic. Reichwald and Behrbohm (1983) also recognize this dimension, though, with other values (operational, structural, and strategic). Another example of different opinions about a dimension concerns the source of flexibility which can be external or internal. Although used by several authors (see, e.g., Sagasti 1970; Eppink 1978; Reichwald and Behrbohm 1983), they have different interpretations of the dimension. For instance, the internal-external dichotomy of Sagasti (1970) refers to the location of the disturbance that causes a flexible behaviour, while Eppink (1978) discerns between internal and external behaviour (active or passive, respectively).

The limitations of these contributions may be traced to a lack of a fundamental theoretical basis. The collection of dimensions seems to be scraped together without much regard to their origins and relationships.

A statement of the underlying principles is therefore in order. These are based on a liguistic analysis which seeks to create a knowledge representation language (the framework) to capture knowledge about information systems flexibility. This language also provides a basis for many of the contributions described above. Because the knowledge representation language is always grounded in natural language (see Weigand 1989), its essentials are based on grammar theory. There is a world of grammar theories available to choose from (see, e.g., Winograd 1983). They all have in common the use of sets of *semantic primitives* that serve as bridges between the abstract level of knowledge and the representation level of symbols (see, e.g., Newell 1982). These have a certain correspondence to meaning, but, because this relationship is far from simple, the variety of proposed sets of primitives is huge. Well known examples are Systemic Grammar (see Halliday 1961), Case Grammar (see Fillmore 1968), and Conceptual Dependency Grammar (see Schank 1975). All these grammars are open-ended proposals: there is no overall theory on which they can be based. In fact, finding a minimal set of semantic primitives is an empirical issue (see Weigand 1989). Consequently, the knowledge representation language (i.e., the conceptual framework) to be created is also open-ended and requires further empirical confirmation (a matter for a separate paper). The literature on grammars can only serve to choose a suitable point of view and create a plausible set of primitives for the framework. This task is dealt with in the next section.

4. ESTABLISHING THE CONCEPTUAL FRAMEWORK

Each line of reasoning has its basic assumptions. So has the following. Three assumptions have been made. The first and most basic one is that the understanding of information systems flexibility can be increased by observing what happens with it in different situations and searching for patterns. Actions, properties and perceptions play different roles in these situations, like the roles of a scenario. Fillmore (1968) calls such roles *case roles*, i.e., "semantically relevant syntactic relationships." Schank (1975) speaks of *conceptual cases*. In this paper the term *role* is adopted to emphasize the scenario metaphor.

The second assumption is that the roles of participant entities in a flexibility scenario can be discovered by asking simple *questions* about the scenario. This has also been suggested by Minsky (1975). The framework to be presented is based on questions about what happens in situations where information systems flexibility is at stake. Questions are also vehicles to find, for each role, a set of *fillers* - conceptual values of roles. For example, as will be seen, the values chosen for the Method role will be Parameters, Library and Development. Roles and fillers facilitate the classification and measurement of a particular scenario. Winograd (1983) calls systems like these *choice systems*.

The third assumption is that actions, properties and perceptions in the realm of information systems flexibility can be separated into two groups, i.e., those concerned with building flexibility and those concerned with mobilizing the phenomenon. The idea behind this is that flexibility has the general character of a potential or ability that can either be built or mobilized. The assumption is necessary because, as will be seen, the observed flexibility potentials focus on different stages of the informations system's life cycle. Without the assumption the questions put forward to capture knowledge about the flexibility phenomenon tend to be open to alternative interpretations.

The remainder of this section first concentrates on the questions and roles concerned with the mobilization of information systems flexibility. The discussion about building flexibility is deferred to the end of the section. Figure 2 depicts a collection of questions and indicates for each question a role that can be deduced from it. The questions chosen are the basic ones for characterizing a phenomenon (when, who, where, how, what). There is no guarantee however that they are adequate, only a conjecture that they are. The roles associated with them have a ground in the grammar literature and have been evaluated on the basis of the frequency with which they are mentioned in the literature concerning organizational flexibility and their potential contribution to information systems flexibility. Some of these references are also listed in Figure 2 for illustrative purposes. The five structural patterns of flexibility which are mentioned in the figure are clarified successively.

Questions about flexibility	Roles	References
When is it mobilized?	Inception	Schank(1975), Eppink(1978), Reichwald(1983)
Who/what does it mobilize?	Agent	Winograd(1983), Ackoff(1971), Reichwald(1983)
Where is it mobilized?	Process	Schank(1975), Ives(1980), Rochester(1989)
How is it mobilized?	Method	Schank(1975), Kickert(1982), Reichwald(1983)
What does the mobilization affect?	Recipient	Sagasti(1970), Ackoff(1971), Schank(1975)

Figure 2. Questions, Roles and References

1. The Inception Role

The first role, Inception, indicates the point of mobilization of the flexibility potential. Reichwald and Behrbohm (1983) call it "Einsatzzeitpunkt," others call it Time (see, e.g., Schank 1975). Three points of time are of relevance, i.e., time of disturbance, time of counter-measure, and time of perception. Figure 3 depicts the two possible moments of inception, i.e., Reactive and Anticipative.

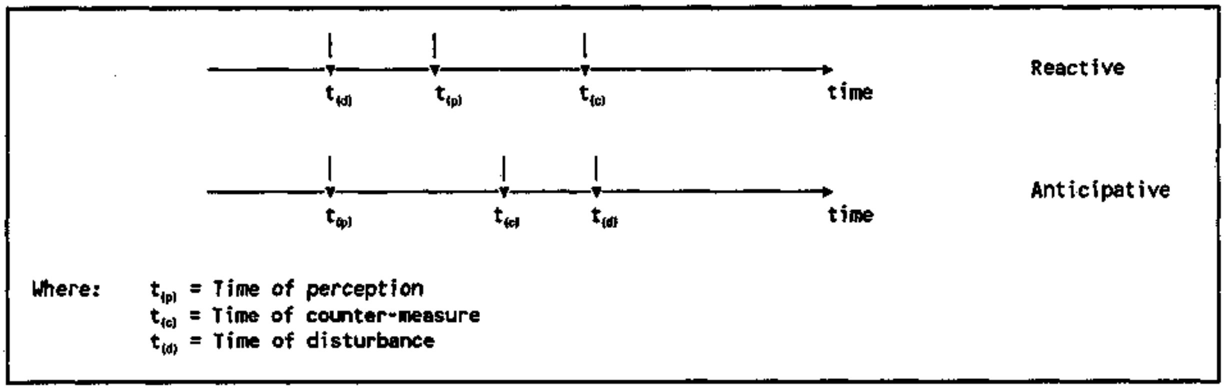

Figure 3. Inception Role

The Inception role is reactive when the time of perception is after the time of disturbance. The pace with which the counter-measure is mobilized is important because the inflexibility costs of an information system can increase significantly $[t_{(c)} - t_{(d)}]$. An example of reactive inception is restructuring the database and application program rewriting as a consequence of a change in one of the business rules in the environment of the information system. For instance, an organization initially requires each of its employees to work on only one project at any time but later changes its policy to allow an employee to work on a number of projects simultaneously (see Sockut and Goldberg 1979).

The Inception role is anticipative when the disturbance is perceived before it occurs.

This implies that the change or disturbance can be foreseen to some extent. Although it is almost impossible to predict all future changes, recent studies indicate that a future or flexibility analysis is both feasible and practical (see, e.g., Fitzgerald 1990). The information system is adapted to cope with the change whenever it actually occurs. During its operation, the information system can be temporarily inefficient [$t_{(d)}$ - $t_{(c)}$]. However, the period of inefficiency must be seen as a kind of insurance investment because when the change occurs the system has low or no inflexibility expenditures to adapt to the change. For example, the above change in a business rule can also be countered anticipatively and incorporated in the database structure in advance and, hence, eliminating the need for database restructuring and consequently, application program rewriting (see Date 1990).

2. *The Agent Role*

The second role, which is called Agent, concerns the origin of mobilization of the flexibility potential. It is an important entity both in the grammar literature and in the literature concerning flexibility (see, e.g., Winograd 1983; Sagasti 1970; Ackoff 1971; Reichwald and Behrbohm 1983). In order to characterize information systems flexibility two types of origins are of importance. Either the mobilization of the potential arises from within the information system itself (internal) or from an outside source (external). The values for the Agent role are therefore chosen to be Internal and External.

An example of an internal Agent is the recovery routine of a Database Management System. The information system itself is in the position to identify the disturbance and to mobilize the flexibility potential without interference from its environment. In contrast, reformatting of the physical database structure is an example of an external mobilizer, i.e., the Database Administrator (see Date 1990).

The application of the values internal and external requires a sharp distinction between aspects that belong to an information system and aspects that are part of its environment. Whereas information systems and their environments are objective entities, they are also subjective because the specific formation of elements that constitute them is prescribed by the interests of the observer. Therefore, the same phenomena can be observed differently by various analysts (see, e.g., Ackoff 1971). In the remainder of this paper, a narrow definition of an information system is utilized, i.e., an information system is an automated system which consists of a set of hardware devices, a filled database and a collection of application programs.

Looking ahead to the application of the flexibility framework (see Section 4), one can expect that most Agent roles will be external because the self-adaptation capabilities of information systems are very limited.

3. *The Process Role*

The Process role refers to the stage at which the flexibility potential is applied. It is an orientation role as defined by Schank (1975) and included in the framework because the flexibility claims of various developments in the realm of information systems concentrate on different stages of the systems life cycle. Ives et al. (1980) describe three stages called processes: development, use, and operations, which make up the interactions between an information system and its environments. The development process yields an information system by selecting and applying environmental development resources such as methods and techniques. For example, CASE technology seeks to provide flexibility during the development process by supporting different diagramming techniques, data modelling approaches, etc. (see McClure 1989). The use process focuses on the utilization of the information system by primary users (i.e., decision makers and intermediaries). For example, relational database management systems profess flexibility during the utilization of the information system by means of an ad hoc query facility (see Date 1990). The operations process concerns the physical operation of the information system (e.g., performance, service, etc.). For instance, non-proprietary front-ends emphasize flexibility during the information system's operation by facilitating database management system independence.

4. The Method Role

The Method role emphasizes the procedure that is followed when the flexibility potential is mobilized. It indicates whether flexibility is revealed through the modification of information system parameters, the application of a library with predefined elements, or through the development of new elements. Parameters, library, and development are the instruments of the procedure (see, e.g., Schank 1975; Reichwald and Behrbohm 1983). Several development methods concentrate on designing for change (see Veldwijk 1993; Ligezinski 1988). Consequently, many required changes to the information system can be effected by altering parameters rather than by modifying the database structure or application programs. On the other hand, programming languages, for example, contain libraries of functions that enable the programmer to (re)use these functions during the implementation of application programs. Required changes to information system's elements that are not available in the library or cannot be realized by modifications to parameters have to be developed by the programmer.

5. The Recipient Role

The final role, Recipient, involves the implications of the mobilization of flexibility potential on the elements of the Information System. Support for this role can be found in several grammar proposals (see, e.g., Schank 1975; Weigand 1989) and other literature (see, e.g., Ackoff 1971). Sagasti (1970) distinguishes two aspects that can be affected, i.e., structure and function. In the context of information systems, structure is subdivided into architecture and database structure. Downsizing is an example of an alteration of the architecture of an information system (see Doll and Doll 1992). Database restructuring is an example of modification of the database structure (see Sockut and Goldberg 1979). The functions of a system are a product of the outcomes that define its goals and objectives. In the case of an information system, they are incorporated in the application programs. Functional alterations are therefore changes in the application programs. It should be noted that the boundary between database structure and application programs is rather indistinct and depends on existing conventions (see Boogaard and Veldwijk 1993; Kent 1989). Consequently, the aspects that belong to the realm of the database structure or to the domain of application programs differ between information systems.

These five roles of flexibility and their accompanying values result in the following conceptual information systems flexibility framework (see Figure 4).

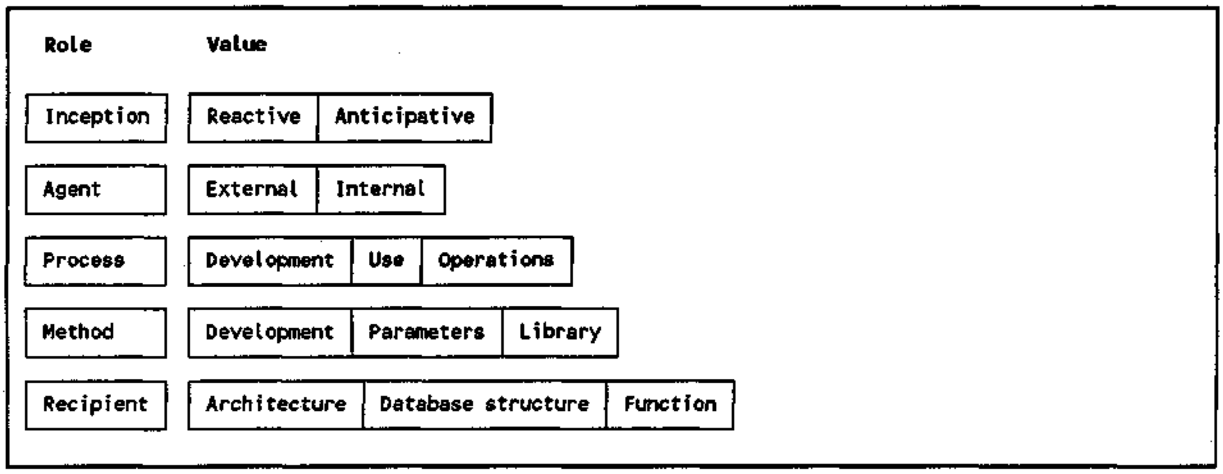

Figure 4. Information Systems Flexibility Framework

A major difference between the framework approach and the two other approaches mentioned in Section 2 (i.e. general comprehensive definitions and specific contextual definitions) is that the framework does not provide a definition of information systems flexibility. It is a classification mechanism that enables to recognize different kinds of flexibility. In fact, the framework may lead to a number of specific contextual definitions, one for each class. For instance, the definition of Veldwijk (1993) mentioned in Section 2 originates in the technique to avoid hard-coding as much as possible by shifting functional aspects to the data structure and data instance realm and by applying meta data structures. The roles and values for the mobilization of flexibility are in this case: Agent = External, Process = Use, Recipient = Database structure, Method = Parameters, and Inception = Reactive.

The framework also facilitates a comparison and evaluation of different classes because the values of some of the roles in the framework (e.g., Agent and Method) are a measure for the *effort* needed to mobilize a flexibility potential. For example, from a flexibility point of view an information system that can be altered by means of a library is more flexible than a system that can only be changed through development. Likewise, an internal agent is preferable over an external agent, because external means costly human labour while internal implies that the work is done by a system. This measure of effort is related to the time dimension given by Zelenovic (1982). His time dimension is a yardstick of the time needed to perform the mobilization of a flexibility potential (not to be confused with the inception time). Effort has not been included into the framework as a separate role because the framework already covers it implicitly.

The discussion so far has deliberately focused on the mobilization of information systems flexibility, because the building or creation of a flexibility potential can be explained in terms of the roles and values found for mobilization. As a matter of fact, the framework for building flexibility can be constructed as subset of the mobilization framework. Successively, the five roles have the following values in case of building: (1) Agent is always human, at least at the current state-of-the-art. Therefore, its value is External; (2) Process is always Development; (3) Recipient has the same value as during mobilization of the potential that is built; (4) Method also has only one value, i.e., Development; and finally (5) Inception can only be Anticipative, because building of flexibility anticipates future changes (while its mobilization can either be reactive or anticipative). The framework therefore serves as a vehicle to represent both the mobilization as well as the building of information systems flexibility. However, as has been said earlier, it is at present no more than a basis for discussion, an initial proposal that requires further empirical confirmation. This is beyond the scope of the current paper, but the following section discusses the potential practicability of the framework.

5. APPLICATION OF THE FRAMEWORK

A number of approaches to information systems analysis, design and implementation claim to introduce or increase flexibility. However, it has been made clear in the introduction section that they mean different things and have different goals with it. In this section, three approaches are discussed and positioned on the basis of the conceptual framework. Successively, object orientation, CASE technology, and relational systems will be considered because of their current significance in theory and practice. The flexibility claims of the three approaches are evaluated on the basis of theoretical descriptions, rather than the application and practical realization of their claims.

5.1 Object Orientation

The perspective of object orientation is based on concepts derived from object oriented programming languages which in turn evolved from conceptual work by programming language theorists on abstract data types (see Heintz 1991; Meyer 1988). An object is defined as an entity that is described by a collection of attributes which represent the object's state and procedures called methods that describe the object's behaviour. Instead of calling functions, communications between objects is based on passing messages. Object orientation, whether it is applied to analysis, design, or programming, is often defined by four core concepts, i.e., encapsulation, classification, inheritance, and polymorphism (see, e.g., Pawson 1991). *Encapsulation* entails combining both code (methods) and data (attributes) within self-contained objects. Data within objects can only be accessed and changed through the methods in the same object. *Classification* means that each object is an instance of a class (object type). A class can be regarded as a template that defines both the data held within objects of that class, and the methods that can operate on them. The third concept, *inheritance*, is the capability of an object automatically to inherit attribute definitions and methods from higher-level classes. Subclasses inherit superclass attributes and methods, but can also modify or extend this definition through the definition of locally owned attributes and methods. Finally, *polymorphism* denotes that different subclasses can interpret the same message in different ways. Consequently, polymorphism reduces the amount that programmers need to know about the different characteristics of various classes. The coding of the (general) method that interprets the message will be specific for each subclass.

To conclude, object orientation claims flexibility by facilitating a modelling approach that closely reflects human conceptualization of the system domain and thus enhances maintainability. Furthermore, object orientation focuses on reusability and extendability which is realized by means of inheritance (generic code used by subclasses), encapsulation (modularized to a certain point where it is easy to extract and modify), and minimization of object interdependencies.

Object orientation can be evaluated on the basis of the roles and values of the conceptual framework for information systems flexibility as follows. The flexibility potential of object orientation must be mobilized by an external Agent. Object orientation concentrates on the development Process by building a comprehensive library of general-purpose object classes, which enables the creation of new information systems from existing objects. Object orientation can therefore dramatically reduce the effort of new systems development. Establishing an appropriate class hierarchy is crucial to achieve all the benefits of object orientation. Yet the process of identifying an initial class hierarchy is far from intuitive and there are only a few formal analysis methods available (see Pawson 1991). The Recipient role is dual valued in this case because object orientation combines data structures and functions into objects. The benefits of object orientation primarily derive from the reusability of objects. Therefore, object orientation is focused on the Library value of the Method role. Whenever a situation occurs that does not quite fit in the current class hierarchy, it is simple to create a subclass. In addition, required objects that do not fit in the class hierarchy at all need to be developed separately. The Inception role for object orientation is reactive. Although establishing of a class hierarchy can be considered anticipative (see the discussion at the end of the previous section), the counter-measure takes place after the disturbance. The pace with which the counter-measure can be mobilized and executed is enhanced by reusability.

5.2 CASE Technology

The basic idea behind CASE technology is to supply a collection of well-integrated, labour-saving tools linking and automating all phases of the systems development life cycle (see McClure 1989). CASE technology is a collective noun for different tools that support (a part of) the system development life cycle. Normally, three kinds of CASE tools are distinguished, i.e., upper CASE tools, lower CASE tools, and integrated CASE tools. The distinction between these types of CASE tools depends on the stages of the systems development process they support. Upper CASE tools primarily support the analysis and functional design stages of the systems development process, whereas lower CASE tools mainly concern the technical design and implementation stages. Integrated CASE tools support all the stages of the systems development life cycle and normally include support for project management. In general, CASE technology focuses on productivity enhancement of the information systems development process through automatic code generation. In order to achieve higher productivity, CASE technology provides among others: several adjustable development methodologies, a repository, screen and report painters, diagramming tools, specification checking tools, code/database generators, re(verse) engineering tools, project management tools, and documentation generators. To conclude, the flexibility claims of CASE technology mainly concern productivity and maintainability. Furthermore, CASE technology include support for several types of diagramming techniques and customizable development methods. Finally, the CASE repository is a mechanism for providing and controlling reusable components. Therefore, CASE technology makes reusability practical (see McClure 1989).

CASE technology can be evaluated in terms of the information systems flexibility framework as follows. The flexibility potential provided by CASE technology is mobilized externally (Agent role). Furthermore, CASE technology focuses on the value Development of the Process role by supporting the overall development process automatically, facilitating several methodologies and diagramming techniques from which the developers can choose, and by enabling prototyping. The effect of CASE technology on values of the Recipient role is the design of the database and the application programs. Consequently, required changes in the database structure or application programs can be easily located. As CASE technology primarily concentrates on productivity and maintainability, the impact on the value Development of the Method role is quite large. In addition, the Library value of the Method role is also supported to some extent by providing reusability. The value of the Inception Feature for CASE technology is Reactive. Again because of its emphasis on productivity, the pace with which the counter-measure can be mobilized and executed is enhanced by applying CASE technology.

5.3 Relational Systems

The relational approach represents the dominant trend in the database management systems (DBMS) marketplace today (see Date 1990). These DBMSs are based on the relational model which was introduced around 1970 (see Codd 1970). Since then the relational model has been refined by its leading advocates Codd and Date (see, e.g., Codd 1970; Codd 1990; Date 1990). Briefly, a relational system is a system in which: (1) all data is perceived by the user as tables and nothing but tables, and (2) the operators that are at the user's disposal are operators that generate new tables from old. The relational model represents a major advance toward the goal of data independence and data consistency. The flexibility claims of relational systems mainly concern the provision of data independence and productivity improvement by means of applying high-level non-procedural relational languages. Data independence can be defined as the immunity of application programs and terminal activities to change in storage structure and access technique. Generally, within relational systems two kinds of data independence are considered, i.e., physical data independence (users and user programs are independent of the physical structure of the database), and logical data independence (users and user programs are independent of the logical structure of the database). The advantage of increased productivity can be traced to several factors but primarily corresponds to the simplicity and power of relational languages, e.g., SQL (see Codd 1990). Furthermore, RDBMSs contain several facilities to control the data, for example, recovery and concurrence controls.

The evaluation of relational systems on the basis of the roles and values of the information systems flexibility framework can be described as follows. In relational systems the flexibility potentials are generally mobilized externally. However, several potentials, for instance, the recovery routine and automatic bind process of RDBMSs, are mobilized internally (see Date 1990). Relational systems focus on both the development and use process by increasing productivity and data independence, and facilitating a relational language. The effect of relational systems on the Recipient role is the provision of data independence and a data manipulation language for application programs, and a relational approach and a data definition language for the database structure. The value Architecture is also influenced to some extent by means of distribution independence. Furthermore, relational systems primarily concern the Development value of the Method role. RDBMSs sometimes respond to changes by applying resource library functions. The value of the Inception role for relational systems can be Reactive but also Anticipative. For instance, the provision of logical data independence by applying the view mechanism requires an anticipative inception. Alteration of the physical database structure by means of DDL is an example of reactive behaviour.

5.4 Comparison of the Approaches

The contribution of the three approaches to the various roles of information systems flexibility can be summarized by means of Figure 5. The figure shows only the mobilization of flexibility. Moreover, only those flexibility contributions are indicated in the framework that are emphasized in the theoretical descriptions of the three approaches. The impact of the different approaches on the other values of a role are omitted for simplicity. For instance, each approach asserts to provide portability (Operations value of the Process role) but does not consider it as their main flexibility attribute. The result is a comparison of the three approaches discussed.

	Approaches		
Role	Object orientation	CASE technology	Relational Systems
Inception	Reactive	Reactive	Reactive and Anticipative
Agent	External	External	Generally External limited Internal
Process	Development	Development	Use and Development
Method	Library	Development and Library	Development and limited Library
Recipient	Database structure and Function	Database structure and Function	Architecture, Database structure and Function

Figure 5. Contribution of Approaches to Information Systems Flexibility

Examining the roles and values of Figure 5 shows that, as far as flexibility is concerned, object orientation and CASE technology look very much alike, whereas relational systems deviate. So, the question "Do object oriented supporters and CASE technology supporters mean the same thing if they talk about flexibility?" put forward in the introduction to this paper, may be answered positively, that is, to a certain extent. Between object orientation and CASE, all values correspond, except for the Method role. According to the values of this role, object orientation is more flexible than CASE (that is, as far as mobilization is concerned) because development requires more effort (an implicit role, see previous section) than library. However, CASE tends to emphasize reusability (i.e., shifts towards library) more and more. One should keep in mind though, that this advantage very much depends on the availability of an adequate library that has to be developed too.

Relational systems differ from the other two systems. Figure 5 demonstrates that relational systems show flexibility in other areas than object oriented systems and CASE systems. For instance, relational systems offer flexibility during usage, whereas neither of the other two do. Furthermore, relational systems explicitly deal with distribution independence as a flexibility aid, which is not the case with the other two. Moreover, logical data independence facilitates anticipative inception whereas this is not the case with the other two.

The differences between the characteristics of flexibility of object oriented systems and relational systems might explain the current interest in so-called *object oriented databases* (see Stein and Maier 1988). As far as the flexibility framework is concerned, object orientated systems and the relational systems are complementary. Among other things, object oriented databases combine the flexibility advantages of object orientation and relational databases. Supporters of both camps, therefore, could learn from the framework and the observations made with it.

6. CONCLUDING REMARKS

Information systems flexibility is a phenomenon that seems to have many dimensions. Traditional definitions of it however tend to either cover just a small part of these dimensions (narrow definitions) or stay at too high a level to be meaningful (broad definitions). This paper seeks to introduce a classification framework that can be used to increase the understanding of the dimensions of information systems flexibility ahead of possible definitions. In other words, definitions might be deduced from the application of the framework.

Although many studies on flexibility in related areas accept that it is multidimensional, discussions about the underlying fundamentals and assumptions are very rare. This paper pays specific attention to both the fundamentals and the assumptions underlying the framework. The dimensions of flexibility have been recognized as the semantic primitives of a linguistic grammar. These primitives represent the roles played in a scenario in which flexibility is at stake. In other words, the understanding of flexibility goes through looking at what happens with it. It was concluded that the collection of semantic primitives cannot be found on the basis of some theory. Instead, a minimum and adequate set of primitives for information systems flexibility has to be established empirically. The framework presented in this paper is therefore an open-ended proposal, a basis for discussion. Nevertheless, the proposed set of primitives is plausible for two reasons: the primitives are selected on the basis of fundamental questions which makes them likely to be relevant, and they also receive much support from both the grammar and the flexibility literature. The primitives are therefore conceived to be the underlying basis of the flexibility framework.

The construction of the framework and its application to three important information technologies, uncovered that the framework not only enables classification of types of flexibility but also facilitates comparison of different types in terms of their flexibility. The effort needed to act in a flexible manner (an implicit feature of the framework) can be used as the measure to compare different manifestations of information systems flexibility. The comparison in Section 4, for instance, showed that object orientation and relational systems complement each other in terms of flexibility. This suggests that it might be useful to classify and, where possible, compare other approaches, issues, and systems in the information systems realm than those treated in Section 4. However, the measurement of the degree of flexibility and also the operationalization of flexibility require further research. Another interesting research issue may be a quantitative analysis of the degree of flexibility of individual information systems based on the flexibility framework.

ACKNOWLEDGMENTS

The authors would like to thank the members of the Department of Information Systems at the Vrije Universiteit Amsterdam and René Veldwijk, and Matthew Jones (Cambridge University, UK) in particular for their contribution to this paper during the discussions concerning the flexibility of information systems.

REFERENCES

1. Ackoff, R.L. (1967), "Management Misinformation Systems," *Management Science*, 14, 4, B147-B156.
2. Ackoff, R.L. (1971), "Towards a System of Systems Concepts," *Management Science*, 17, 11, pp. 661-671.
3. Ackoff, R.L. (1977), "Towards Flexible Organizations: A Multidimensional Design," *OMEGA The International Journal of Management Science*, 5, 6, pp. 649-662.
4. Allen, B.R., and Boynton, A.C. (1991), "Information Architecture: In Search of Efficient Flexibility," *MIS Quarterly*, December 1991, pp. 435-445.
5. Ansoff, H.I. (1978), "The Changing Shape of the Strategic Problem," in Schendel, D.E., and Hofer, C.W. (eds.), *Strategic Management: A New View of Business Policy and Planning*, Boston: Little, Brown & Co.
6. Boehm, B.W., Brown, J.R., Kaspar, H., Lipow, M., MacLeod, G.J., and Merritt, M.J. (1978), *Characteristics of Software Quality*, TRW Series of Software Technology 1, Amsterdam: North Holland Publishing Company.
7. Boogaard, M., and Veldwijk, R.J. (1993), "Blurring Boundaries to Improve Information System Flexibility," *Proceedings of the Third International Conference for Young Computer Scientists*, Beijing/China, July 15-17 1993, pp. 5.75-5.78.
8 Cameron, K.S. (1986), "Effectiveness as Paradox," *Management Science*, 32, pp. 539-553.
9. Codd, E.F. (1970), "A Relational Model for Large Shared Data Banks," *Communications of the ACM*, 13, 6, pp. 377-387.
10. Codd, E.F. (1990), *The Relational Model for Database Management, Version 2*, Reading, Massachusetts: Addison-Wesley Publishing Company.
11. Crowe, T.J. (1992), "Integration is Not Synonymous with Flexibility," *International Journal of*

Operations and Production Management, 12, 10, pp. 26-33.

12. Darnton, G., and Giacoletto, S. (1989), "Towards an Enterprise Information Model," *DEC*, Geneva/Switzerland, GD&SG.
13. Date, C.J. (1990), *An Introduction to Database Systems, Volume 1*, 5th edition, Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.
14. Doll, M.W., and Doll, W.J. (1992), "Downsizing at CBS/FOX Video," *Information and Management*, May 1992, pp. 123-139.
15. Dosi, G. (1982), "Technological Paradigms and Technological Trajectories," *Research Policy*, 11, pp. 147-162.
16. Edquist, C., and Jacobsson, S. (1988), *Flexible Automation: The Global Diffusion of New Technology in the Engineering Industry*, Oxford: Basil Blackwell, Inc.
17. Eppink, D.J. (1978), *Managing the Unforeseen: A Study of Flexibility*, Ph.D. Thesis, Ermelo: Administratief Centrum.
18. Fillmore, C. (1968), "The Case for Case," in Bach, E., and, Harms, R.T. (eds.), *Universals in Linguistic Theory*, Chicago: Holt, Rinehart & Winston.
19. Fitzgerald, G. (1990), "Achieving Flexible Information Systems: The Case of Improved Analysis," *Journal of Information Technology*, 5, pp. 5-11.
20. Frazelle, E.H. (1986), "Flexibility: A Strategic Response In Changing Times," *Industrial Engineering*, March 1986, pp. 17-20.
21. Geraerds, W.M.J., and Igel, M. (1989), *Flexibility in Logistics (Flexibiliteit in Logistiek)*, Alphen a/d Rijn: Samsom BedrijfsInformatie. (In Dutch.)
22. Gerwin, D., and Leung, T.K. (1980), "The Organizational Impacts of Flexible Manufacturing Systems: Some Initial Findings," *Human Systems Management*, 1, pp. 237-246.
23. Halliday, M.A.K. (1961), "Categories of the Theory of Grammar," *Word*, 17, pp. 241-292.
24. Heintz, T.J. (1991), "An Object-oriented Approach to Planning and Managing Software Development Projects," *Information and Management*, 20, 4, pp. 281-293.
25. Ives, B., Hamilton, S., and Davis, G.B. (1980), "A Framework for Research in Computer-Based Management IT," *Management Science*, 26, 9, pp. 910-934.
26. Kent, W. (1989), "The Leading Edge of Database Technology," in Falkenberg, E.D., and Lindgreen, P. (eds.), *IT Concepts: An In-Depth Analysis*, Amsterdam: North-Holland.
27. Kickert, W.J.M. (1982), "The Magic Word Flexibility (Het toverwoord flexibiliteit)," *Bestuurswetenschappen*, 7, pp. 476-492. (In Dutch.)
28. Land, F. (1982), "Adapting to Changing User Requirements," *Information and Management*, 5, pp. 59-75.
29. Ligezinski, P. (1988), "Zero-Maintenance Systems," second place winner of the *International Paper Competition of the Society of Information Management*.
30. McClure, C. (1989), *CASE is Software Automation*, Englewood Cliffs, New Jersey: Prentice-Hall, Inc.
31. Meyer, B. (1988), *Object-oriented Software Construction*, New York: Prentice-Hall International, Ltd.
32. Minsky, M. (1975), "A Framework for Representing Knowledge," in Winston, P. (ed.), *The Psychology of Computer Vision*, New York: McGraw-Hill.
33. Nelson, R.R., and Winter, S.G. (1977), "In Search of a Useful Theory of Innovation," *Research Policy*, 6, pp. 36-76.
34. Newell, A. (1982), "The Knowledge Level," *Artificial Intelligence*, 18, 1, pp. 87-127.
35. Pawson, R. (1991), "Object Orientation," *Butler Cox Foundation Research Report*, September 1991.
36. Perez, C. (1985), "Microelectronics, Long Waves and World Structural Change: New Perspectives for Developing Countries," *World Development*, 13, 3, pp. 441-463.
37. Piore, M., and Sabel, C. (1984), "Why Companies Might Be Moving Steadily Towards Specialization and Flexibility," *International Management*, October 1984, pp. 97-100.
38. Pirsig, R.M. (1981), *Zen and the Art of Motorcycle Maintenance*, New York: Bantam Books.
39. Reichwald, R., and Behrbohm, P. (1983), "Flexibility as Property of Production Systems (Flexibilität als Eigenschaft produktionwirtschaftlicher Systeme)," *Zeitschrift für Betriebswirtschaft*, 53, 9, pp. 831-853. (In German.)

40. Rochester, J.B. (1989), "Building More Flexible Systems," *I/S Analyzer*, 27, 10, pp. 1-12.
41. Sagasti, F. (1970), "A Conceptual and Taxonomic Framework for the Analysis of Adaptive Behavior," *General Systems*, 15, pp. 151-160.
42. Schank, R.C. (1975), *Conceptual Information Processing*, Amsterdam: North Holland.
43. Scott, B.W. (1965), *Long-Range Planning in American Industry*, New York: American Management Association.
44. Sockut, G.H., and Goldberg, R.P. (1979), "Database Reorganization: Principles and Practice," *ACM Computing Surveys*, 11, 4, pp. 371-395.
45. Spoor, E.R.K., Boogaard, M., and Veldwijk, R.J. (1992), "The Transformation of Extended Entity Relationship Generalisation Hierarchies into Tables and Meta Tables," *European Journal of Information Systems*, 1, 5, pp. 333-340.
46. Standing, G. (1991), *In Search of Flexibility: The New Soviet Labour Market*, Geneva: International Labour Office.
47. Stein, J., and Maier, D. (1988), "Concepts in Object-Oriented Data Management," *Database Programming and Design*, 1, 4.
48. Steers, R.M. (1975), "Problems in the Measurement of Organizational Effectiveness," *Administrative Science Quarterly*, December 1975, 20, pp. 546-558.
49. Volberda, H.W. (1992), *Organizational Flexibility: Change and Preservation*, Groningen: Wolters-Noordhoff.
50. Veldwijk, R.J. (1993), *Introspective Systems Design: Exploring the Self-Referential Capabilities of the Relational Model*, Ph.D. Thesis, Enschede: Copyprint 2000.
51. Weigand, H. (1989), *Linguistically Motivated Principles of Knowledge Base Systems*, Ph.D. Thesis, Dordrecht: ICG Printing.
52. Winograd, T. (1983), *Language as a Cognitive Process*, Reading, Massachusetts: Addison-Wesley Publishing Company, Inc.
53. Zelenovic, D.M. (1982), "Flexibility: A Condition for Effective Production Systems," *International Journal of Production Research*, 20, 3, pp. 319-337.